First Named Inventor	Ghose, et al.	**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE** In Re Application of: Ghose et al.
Serial No.	10/695,889	
Filing Date	10/20/2003	
Group Art Unit	2114	
Examiner Name	Gabriel L. Chu	
Confirmation No.	7789	
Docket No.	00121-000700000	
Title: FAILURE ANALYSIS METHOD AND SYSTEM FOR STORAGE AREA NETWORKS		

VIA EFS

APPEAL BRIEF

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

As required under 37 C.F.R. § 41.37(a), this Appeal Brief is being submitted in furtherance of the Notice of Appeal filed on February 20, 2008. The fees required under 37 C.F.R. § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF. This Appeal Brief contains items under the following headings:

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The present application has been assigned in an assignment recorded at Reel 015963, Frame 0680 to NetApp, Inc., a Corporation established under the laws of the State of Delaware and having a principal place of business at 495 East Java Drive, Sunnyvale, CA 94089, U.S.A..

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any other related appeals or interferences that may directly affect or be directly affected by or have a bearing on the Board's decision in the present appeal.

III. STATUS OF THE CLAIMS

Claim 1-4, 6-11, 18-21, 23-26, 33 stand rejected under 35 U.S.C. § 103(a) over U.S. Patent No. 5,666, 481 to *Lewis* (hereinafter "Lewis") in view of U.S. Patent Publication 2002/0019922 to *Reuter* et al. (hereinafter "Reuter") and U.S. Patent No. 6,629, 266 to *Harper* (hereinafter "Harper").

Claim 12-16, 27-32, 34, 36 stand rejected under 35 U.S.C. § 103(a) over Lewis in view of Reuter and further in view of U.S. Patent 6,336,139 to *Feridun* (hereinafter "Feridun") and U.S. Patent 6,446,218 to *D'Souza* (hereinafter "D'Souza").

Claim 35 stands rejected under 35 U.S.C. § 103(a) over Lewis in view of Reuter, the term "threshold" as defined by the IEEE and "graphical user interface" as defined by the Microsoft Computer Dictionary (hereinafter MSCD) and U.S. Patent 6,629,266 to *Harper et al.* (the "Harper" patent).

Appellant appeals the rejection of all of the pending claims 1-4, 6-16, 18-21, 23-36 which are set forth in the attached Appendix A.

IV. STATUS OF AMENDMENTS

The amendments to the claims have all been entered and Appellant is unaware of any amendments filed after the Final Office Action mailed 11/20/2007 which finally rejected claims 1-4, 6-16, 18-21, 23-36.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Claims 1-4, 6-16, 18-21, 23-36 are directed to an error and failure analysis and management method and system for a storage area network (SAN). There are many features in this method and system not previously available in the storage area arena. For example, the error and failure analysis is performed on a centralized failure analysis module even though the errors or other alerts come from distributed storage controllers and storage systems.

Different errors and failures occurring on many different subsystems in the SAN or on the storage controllers can be analyzed more readily on a centralized failure analysis module. This information can be used to rapidly identify failing systems and take actions to ameliorate the damage or loss of data. For example, the centralized failure analysis module can direct various distributed storage controllers performing storage virtualization to relocate data from failing storage systems to more reliable storage systems. Many other types of recovery operations can take place by way of the centralized failure analysis module.

Further, another feature of the method and system provides opportunities for backup systems to take over processing in the event a centralized failure analysis module is abruptly shutdown or fails. In a SAN having a distributed set of storage controllers, one storage controller can be designated as housing the primary failure analysis module while other storage controllers can be designated to hold the secondary or tertiary failure

analysis modules in the event a storage controller or primary failure analysis module becomes unavailable or goes down.

The method and system as claimed uses rules to govern the detection and management of errors and failures in the storage area network. Rule-driven management of the SAN associates customized rules with error actions through a non-programmatic interface. These rules may detect one or more predetermined error events and invoke one or more error actions. For example, a SAN administrator can setup the error management system of the present invention through a graphical user interface (GUI). The GUI interfaces with object-oriented methods and instances according to the configuration information thereby making the system easier to use and deploy. Further, rules can be developed incrementally and over time as problems on the SAN arise and are understood without having to re-code or throw away previous work setting up the error and failure analysis and management. This allows implementations of the present invention to grow and change with changing use of the SAN.

Representative claim 1 (Appellant's Specification, Figures 2, 3, 4, 5 and paragraphs [0014], [0030], [0031], [0032], [0033], [0052]) describes a method for configuring a storage virtualization controller to manage errors in a storage area network. To start, the configuration initializes a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module. This arrangement facilitates high-availability and redundancy by having at least two failure analysis modules configured to work together in a fail-over modality (Appellant's Specification, Figure 2, paragraphs [0014], [0031], [0052]).

Next, the configuration method identifies one or more predetermined error actions and one or more error events associated with the storage area network. (Appellant's Specification, Figure 4, paragraph [0029]; Figure 5 paragraph [0031]). As illustrated in Figure 4 and described in the specification, the failure analysis module collects a variety of error events over time and then enumerates them with a corresponding set of error codes. For example, a fan failed event may be converted into an error code enumerated as E1 and an over temperature event may be converted into an error code enumerated as E2. Timing information is also recorded in association with the occurance of the error events

to provide further context to the occurance of one event and its relationship to another event. (*Id.*)

Further, the configuration method in Claim 1 specifies, according to one or more rules, an error pattern based upon a combination of one or more error events in the storage area network (Appellant's Specification, Figure 4, paragraph [0030]; Figure 5 paragraph [0032]). Each of the error patterns includes timing information about the error events as well as the sequencing or grouping of the error events. In one implementation, the error pattern may specify that the error events occur in a particular sequence and during specific time intervals, or alternatively the error pattern may accept error events that occur in any order within a particular time interval. (*Id.*)

This configuration method then associates an error action to perform according to the one or more rules and in response to receiving the combination of one or more error events of the error pattern (Appellant's Specification, Figure 4, paragraph [0030]; Figure 5 paragraph [0033]) The rules describe error actions to perform and operations to accommodate or counteract the effects of the one or more error events occurring on the storage area network (*Id.*) As previously described, this method and related systems allows an administrator to quickly combine error events into error rules and associate them with error actions to perform by way of the storage virtualization controller in the storage area network.

Representative claim 12 (Appellant's Specification, Figures 2, 3A, 3B, 4 and paragraphs [0014], [0031], [0034], [0035], [0036], [0037],[0038], [0052]) describes managing the occurrence of errors generated in a storage area network having a rule-driven configuration. To maintain high-availability, the error manager initializes a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module. As previously described, this arrangement facilitates high-availability and redundancy by having at least two failure analysis modules running and configured to work together in a fail-over modality (Appellant's Specification, Figure 2, paragraphs [0014], [0031], [0052])

The error manager monitors the condition of various components on the storage area network and then generates error events in response. (Appellant's Specification,

Figure 3A and 3B, Figure 4, paragraphs [0034]). Different monitor modules from the primary failure analysis module track a particular condition occurring on the components in the storage area network or within a storage virtualization controller. For example, a temperature monitor module may monitor for an over-temperature condition in the storage virtualization controller and then convert the error event into an error event codes readily processed by the failure analysis module. (*Id.*) Not just one, but multiple error events may be received by the error manager over a time interval for analysis in the primary failure analysis module (Appellant's Specification, Figure 3A and 3B, Figure 4, paragraphs [0035]). In some cases, a single error may not be sufficient to invoke an error action unless combined with other types of errors. Alternatively, the rules may indicate that some errors events may be severe enough (i.e., over-temperature conditions) to warrant immediate execution of error actions and recovery procedures that shutdown one or more components in the storage area network (*Id.*).

The primary analysis module then uses one or more rules to compare a temporal arrangement of the error events received against a set of error patterns loaded in the primary failure analysis module (Appellant's Specification, Figure 3A and 3B, Figure 4, paragraphs [0036]). The error events can be a combination of system error events and input-output error events and the temporal information can be either the relative timing of the events or an absolute measurement of the timing relative to a clock. Timing and sequencing of these error events are important to determine if the error events warrant taking an error action or other corrective measures (*Id.*).

Depending on the actual error events received, the error manager uses the one or more rules to identify the error pattern from the set of error patterns and error action corresponding to the error pattern to perform in response to the comparison (Appellant's Specification, Figure 3A and 3B, Figure 4, paragraph [0037]) In one implementation, the error patterns are determined in advance and loaded into the failure analysis module during the configuration operations previously described. In most cases, an administrator or operator familiar with operation of the storage area network creates rules that define the error patterns based upon their experience and observation of error events over time (*Id.*). To avert problems on the storage area network, rules can direct the storage virtualization controller to perform a variety of actions to mitigate or recover from certain

errors. For example, rules may instruct the storage virtualization controller to migrate data from a storage element generating error events to other more reliable areas of the storage network not experiencing the error events or failures.

Depending on the situation, alternate error actions may direct the storage virtualization controller to migrate data to more reliable RAID type devices rather than a JBOD (just a bunch of disks) device or other less reliable storage options. (Appellant's Specification, Figure 3A and 3B, Figure 4, paragraphs [0038])

The following listing provides a reference between each independent claim and some, but not necessarily all, paragraphs and figures in the specification:

1. Independent Claims 1, 18, 33, 35: (Appellant's Specification, Figures 2, 3, 4, 5 and paragraphs [0014], [0030], [0031], [0032], [0033], [0052])

2. Independent Claims 12, 27, 34, : (Appellant's Specification, Figures 2, 3A, 3B, 4 and paragraphs [0014], [0031], [0034], [0035], [0036], [0037],[0038], [0052])

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. WHETHER CLAIM 1, CLAIM 18 AND CLAIM 33 IS NONOBVIOUS OVER LEWIS IN VIEW OF REUTER AND HARPER.

B. WHETHER CLAIM 12, CLAIM 27, CLAIM 34 AND CLAIM 36 ARE NONOBVIOUS OVER LEWIS IN VIEW OF REUTER AND FURTHER IN VIEW OF FERIDUN AND D'SOUZA.

C. WHETHER CLAIM 35 IS NONOBVIOUS OVER LEWIS IN VIEW OF REUTER AND THE DEFINITION OF "THRESHOLD" BY THE IEEE AND "GRAPHICAL USER INTERFACE" BY THE MICROSOFT COMPUTER DICTIONARY (MSCD)AND FURTHER IN VIEW OF HARPER.

VII. ARGUMENT

Appellant respectfully traverses the outstanding rejections of the pending claims, and requests that the Board overturn the outstanding rejections in light of the remarks contained herein. The claims do not stand or fall together. In fact, Appellant presents separate arguments for various independent and dependent claims. Each of these arguments is presented with separate headings and sub-headings, as required by 37 C.F.R. § 41.37(c)(1)(vii).

A. CLAIM 1, CLAIM 18 AND CLAIM 33 ARE NONOBVIOUS OVER LEWIS IN VIEW OF REUTER AND HARPER

The Lewis Reference

Lewis describes a method for managing communication networks using a trouble ticket system (Abstract, Col. 3, lines 36-44). According to Lewis, the term "communication network" describes a digital communication system that not only includes local area networks (LANs) but also includes wide-area networks or WANs.

(Col. 1, lines 10-14.) Indeed, Lewis mentions both a LAN and a WAN as both are examples of communication networks that transmit digital information between computers. (Col. 1, lines 9-13). However, there is no mention of storage area networks (SAN) in Lewis that manage both the storage and transmission of data over a network.

The trouble ticket system in Lewis is driven by a database collection of trouble tickets collected from the past. (Col. 6, lines 39-42). Trouble tickets allow users to enter descriptive information on a network condition but not the actual errors (Col. 1, lines 35-40.) Users can only enter observations or symptoms of a network condition as they appear to the user. The actual errors in the network causing the network condition are not known to the user and not recorded in the trouble ticket system. (Col. 1, lines 61-66.)

In fact, the trouble ticket system in Lewis is driven by the symptoms or observations reported in the past and not by rules created in advance. (Col. 6, 33-50). Lewis is limited to operating with "case based reasoning" (CBR) and explicitly excludes the use of "rule based reasoning" (RBR) that associates error patterns with error actions. *(Id.)* Each trouble ticket in Lewis represents a case that is then used to drive a solution. According to Lewis, initializing systems with rules that associate error patterns with error actions does not work and eventually will fail. (Col. 2, lines 50-67). For at least this reason, Lewis makes a strong argument that "case based reasoning" should be used and, in contrast, "rule based reasoning" involving the use of rules should avoided. (*Id.*) Lewis teaches away from "rule based reasoning."

The Reuter Reference

Reuter concerns the management of certain mapping tables used to virtualize storage area network (SAN) systems. (Paragraph 9 of Reuter). In particular, Reuter is

concerned with improving the performance characteristics associated with storing data and accessing stored data using distributed mapping tables. (Paragraph 13, Paragraph 34). Indeed, Reuter manages page faults typical of virtual storage systems but these are not the errors since page faults occur as a normal part of the operation.

The Harper Reference

Harper concerns increasing software dependability and predicting failure of a software system on a computer (Abstract of Harper). Harper assumes that a software application consuming a larger amount of resources is failing and then switches the job consuming the resources to a secondary node.(Col. 12, lines 61-67). Indeed, the software is not actually failing but consuming too many resources for it to continue operating on that particular node for any length of time (Col. 11, lines 56-63). The application is stopped on the primary node and then restarted on the secondary node (Col. 6, lines 42-54). Essentially, the primary node and secondary node both continue to operate but the software application is merely restarted on one node or the other. (*Id.*)

Harper operates by restarting a poorly written application that consumes too many resources with the hope that the system will not crash due to the overwhelming amount of resources it consumes due a memory leak or other problem (Col. 9, lines 41-58) However, nothing in Harper teaches or suggests that two software applications should be running in parallel as backups to each other. Quite the opposite, Harper's strategy is to never run the application on both machines but to only allow one application to run at a time just in case it is buggy and consumes too many resources (Figure 4, Col. 6, lines 42-54).

a. NO TEACHING, SUGGESTION OR MOTIVATION TO COMBINE LEWIS WITH REUTER AND HARPER

Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. (MPEP 2143.01.) The suggestion or motivation to combine the reference teachings must be based on some logical reason apparent from positive, concrete evidence that justifies the combination. (In re Laskowski, 871 F.2d 115, 10 USPQ 2d 1397 (Fed. Cir. 1989).)

In the instant office action, there is no reason to combine Lewis with Reuter as they deal in non-analogous art. Lewis clearly applies to communication networks and not storage area networks. There is no teaching, suggestion or motivation to apply technology in communication networks to storage area networks as they address distinctly different problems – communication networks deals primarily with data transmission and storage networks with data storage. For at least this reason, it is improper for the office action to combine the networks described in Lewis with the storage systems found in Reuter.

Moreover, two criteria are relevant in determining whether prior art is not analogous and should not be combined: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the art is not within the same field of endeavor, whether it is still reasonably pertinent to the particular problem to be solved. Wang Laboratories, Inc. v. Toshiba, 993 F. 2d 858, 26 USPQ2d 1767 (Fed. Cir 1993).

In this case, Lewis concerns network systems and the management of trouble tickets used for resolving network faults. In the context of this appeal, the network faults described in Lewis are more aptly categorized as "errors" since they result from a failure[1] of hardware or software. When it fails, the resulting communication network does not work as it was designed as it was designed (Col. 1, lines 16-30). Lewis specifically applies only to communication networks and there is no mention of storage systems let alone storage area network (SAN) controllers. Hence, the field of endeavor in Lewis is limited to identifying errors in a conventional communication network and does not concern storage systems.

In contrast, Reuter concerns the management of certain mapping tables used to virtualize storage area network (SAN) systems (Paragraph 9 of Reuter). The trouble ticket management for a communication network found in Lewis is unrelated to virtualization of storage in the SAN of Reuter. In particular, Reuter is concerned with improving the performance characteristics associated with storing data and accessing stored data using distributed mapping tables. (Paragraph 13, Paragraph 34).

There is also no connection between the "errors" being managed in Lewis and the "faults" being processed in Reuter. Indeed, the page faults[2] that occur in Reuter are

---

1 Failure From Wikipedia, the free encyclopedia April 20, 2008

Failure (or flop) in general refers to the state or condition of not meeting a desirable or intended objective. It may be viewed as the opposite of success. Product failure ranges from failure to sell the product to fracture of the product, in the worst cases leading to personal injury, the province of forensic engineering.

2 Page fault as defined in Wikipedia, the free encyclopedia on April 20, 2008
In computer storage technology, a page is a fixed-length block of memory that is used as a unit of transfer between physical memory and external storage like a disk, and a page fault is an interrupt (or exception) to the software raised by the hardware, when a program accesses a page that is mapped in address space, but not loaded in physical memory.

The hardware that detects this situation is the memory management unit in a processor. The exception handling software that handles the page fault is generally part of an operating system. The operating system

expected to occur in the normal course of certain input/output (I/O) operations (Paragraph [0029] of Reuter). Appellants submit that the Examiner has in appropriately equated page faults typical of virtual storage systems with "errors" or "failures" managed by Lewis. (Page 3 of the Office Action mailed May 10, 2007) Even though both Lewis and Reuter use the word "fault", they cover distinctly different areas of technology and the words have distinctly different meanings. For example, Reuter's handling of "faults" relates to distributed agents dealing with mapping table entries for the virtual disk but is not relevant to handling errors. (Paragraph 34 of Reuter.)

Moreover, the trouble ticket system described in Lewis is not reasonably pertinent or related to managing the storage and movement of data in a storage network as described by Reuter. The problem to be solved in Lewis concerns determining a resolution to a network fault that occurs when a system fails or does not work correctly. This is not pertinent to a method of managing the virtual table mappings in Reuter.

In summary, Lewis and Reuter should not be considered analogous art for several reasons: First, they are not from the same field of endeavor as one concerns tracking and resolving network faults caused by the failure of a network or software to function and the other concerns managing the normal occurrence of a page fault in virtualized storage. Second, the method of tracking trouble tickets and resolving network failures for a set of customers (Abstract of Lewis) is clearly unrelated to managing virtual storage and page faults in a storage network (Abstract of Reuter).

The Supreme Court in KSR stated that it is "important [for an examiner] to identify a reason that would have prompted a person of ordinary skill in the relevant field

---

tries to handle the page fault by making the required page accessible at a location in physical memory or kills the program in case it is an illegal access.

to combine the [prior art] elements" in the manner claimed. *KSR Int'l Co. v. Teleflex, Inc.*, No. 04-1350, slip op. at 14 (U.S. April 30, 2007). The Court indicated that there should be an *"explicit"* analysis regarding "whether there was an ***apparent reason*** to combine the known elements ***in the fashion claimed*** by the patent at issue." (*Id.*) (emphasis added). Further, the Court did not reject the use of "teaching, suggestion, or motivation" test as a factor in the obviousness analysis, but rather stated that this test may be indicative of non-obviousness under 35 U.S.C. § 103. Id. at 14-15.

Accordingly, the Examiner has also not provided sufficient reasoning to combine Harper with Lewis and/or Reuter. In the Office Action dated May 10, 2007, the Examiner had only stated that Harper should be combined with Lewis and/or Reuter "for increased software dependability...avoiding the outage". These conclusory statements merely repeat overarching goals in Harper and provide no "explicit" connection to either Lewis and/or Reuter.

As previously described, Lewis is not actually about making a network more dependable but concerns a trouble-ticket management system. Thus, there is no reason to improve the performance of the trouble ticket system in Lewis as the trouble-ticket is tracking the failed network problems but is not the system that is failing. Conversely, Reuter does not even deal with "errors" or system failure at all. Rather, Reuter simply deals with managing page faults as they occur in virtualized storage system and there is nothing in Reuter's about backup systems or the need for such things.

There is no teaching, suggestion or motivation in the prior art to combine the cited art to result in what is claimed, as is required. Consequnetly, Claim 1 is not obvious over Lewis in view of Reuter and further in view of Harper for at least the reasons set forth

above. By analogy independent claims 18 and 33 are also in condition for allowance for at least the same reasons.

b. ALL CLAIM LIMITATIONS ARE NOT TAUGHT OR SUGGESTED BY THE CITED ART.

To establish a prima facie case under 35 U.S.C. § 103(a), the Examiner must determine the "scope and content of the prior art," ascertain the "differences between the prior art and the claims at issue," determine "the level of ordinary skill in the pertinent art," and evaluate evidence of secondary considerations. Graham v. John Deere, 383 U.S. 1, 17, (1966); see also M.P.E.P. § 2141. And, when determining the differences between the applied art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. M.P.E.P. § 2141.02(I).

Indeed, a prima facie case of obviousness may be established under 35 U.S.C. § 103(a) when he prima facie teachings from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art." *In re Bell,* 991 F.2d 781, 782, 26 USPQ2d 1529, 1531 (Fed. Cir. 1993) (quoting *In re Rinehart,* 531 F.2d 1048, 1051, 189 USPQ 143, 147 (CCPA 1976))." Accordingly, the Examiner has the burden under section 103 to establish this prima facie case of obviousness. See *In re Fine*, 837 F.2d 1071, 1074; 5 U.S.P.Q.2D (BNA) 1596; *In re Piasecki, 745 F.2d 1468, 1471-72, 223 USPQ 785, 787-87 (Fed. Cir. 1984).*

Further, the Examiner may only satisfy this burden by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in

the art would lead that individual to combine the relevant teachings of the references. *In re Lalu,* 747 F.2d 703, 705, 223 USPQ 1257, 1258 (Fed. Cir. 1984)*;* see also *Ashland Oil, Inc. v. Delta Resins & Refractories, Inc.,* 776 F.2d 281, 297 n. 24, 227 USPQ 657, 667 n.24 (Fed. Cir. 1985); *ACS Hosp. Sys., Inc. v. Montefiore Hosp.*, 732 F.2d 1572, 1577, 221 USPQ 929, 933 (Fed. Cir. 1984)."

With regards to the prima facie case, it is also a requirement that all the claim limitations must be taught or suggested by the prior art. MPEP § 2143.03 In re Royka , 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson , 424 F.2d 1382, 165 USPQ 494, 496 (CCPA 1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. In re Fine , 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

Contrary to the Examiner's assertion, Lewis alone or in combination with the cited art does not teach each and every feature of the invention as claimed. The Examiner admitted that Lewis does not operate by "initializing a primary failure analysis module for processing error events and error actions" as recited in Claim 1. Nonetheless, to satisfy this limitation in the office action of May 10, 2007, the Examiner merely states "a functioning process must have been initialized at some point" (Office Action mailed May 10, 2007 Page 2, Paragraph 4). While this conclusory statement may appeal to the Examiner's common sense at the time, there is nothing cited as a basis for its supposition. (Feature missing from references; improper to rely on basic knowledge and common sense) *In Re Zurko*, 258 F.3d 1379, 59 USPQ2d 1693 (Fed. Cir. 2001) *on remand from Dickinson v. Zurko,* 527 U.S. 150, 50 USPQ2d 1930 (1999)

It also follows that Lewis does not describe or even suggest "identifying one or more predetermined error actions and one or more error events associated with the storage area network" as recited in claim 1. Unfortunately, the office action dated May 10, 2007 cites Figure 4 from Lewis but does not consider the overall reference as a whole and the actual underlying explaination provided in Lewis with respect to Figure 4 and elsewhere.

In particular, Lewis does not identify error events as recited in claim 1 but instead identifies "cases" using "case-based reasoning" or CBR. (Col. 6, lines 33-39). Lewis makes it a point that a "case" or "scenario" described by a trouble ticket is distinguishable over a combination of error actions and error events – this is a central concept to Lewis. (Col. 3, lines 11-20; Col. 6, lines 33-39.) Lewis very specifically identifies "cases" having "particular, specific, fault resolution scenarios stored in completed trouble tickets" and teaches away from identifying error actions and error events. (*Id.*) By operating on cases, Lewis attempts to resolve faults by operating with a different approach. Specifically, if a trouble-ticket matches exactly, Lewis immediately performs the resolution performed previously in the matching trouble ticket. (Col. 8, lines 55-60) In the event there is not an exact match, Lewis describes adapting the trouble-ticket using a parameterized adaptation module and other approaches. (Col. 8, lines 60-67; Col. 9, lines 1-67).

The Examiner also quickly concludes in the office action mailed May 10, 2007 that Figure 4 also teaches the limitations in claim 1 of "specifying..." and "associating..." but provides no detailed reasoning from Lewis for these conclusions. Unfortunately, even the combination of Lewis with Reuter does not teach or suggest

"specifying, according to one or more rules, an error pattern based upon a combination of one or more error events in the storage area network" and "associating an error action to perform according to the one or more rules and in response to receiving the combination of one or more error events of the error pattern" as recited in claim 1. As previously described, Lewis uses specific trouble tickets and case-based reasoning (CBR) to resolve errors and teaches away from specifying errors patterns based up one or more rules. (Col. 6, lines 33-39.)

Reuter does not concern processing either cased-based reasoning or errors but page faults that occur on SAN. As previously described, there is nothing in Reuter that relates to managing errors let alone the limitations associated with the steps of "specifying, according to one or more rules, an error pattern based upon a combination of one or more error events in the storage area network ..." and "associating an error action to perform according to the one or more rules and in response to receiving the combination of one or more error events of the error pattern" as recited in claim 1. If and when a page fault occurs Reuter will either ignore a storage location or pass the error code to an application (Paragraph 27, lines 1-9 of Reuter.). Nothing in Reuter descirbes performing error actions as recited in claim 1.

Indeed, Lewis combined with Reuter and further with Harper does not teach or suggest "initializing...an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy" as recited in claim 1. As previously described, nothing in Harper teaches or suggests that two software applications should be running as redundant applications or backups to each other. Quite the opposite, Harper's strategy is to never run the application on both

machines but to only allow one application to run at a time just in case it is buggy and consuming too many resources (Figure 4, Col. 6, lines 42-54 of Harper). If an application in Harper appears to be consuming too many resources, the cluster controller terminates the application and then restarts it on another computer (Figure 5, Steps 505, 506, 507 and Col. 8, lines 1-28 of Harper).

For at least these reasons, independent claims 1, 18 and 33 as currently filed are in condition for allowance. Dependent claims 2-4, 6-11, 19-21, 23-26 are allowable independently as well as by virtue of their direct or indirect dependency on claims 1 and 18 respectively.

Accordingly, Appellant respectfully requests that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to claims 1, 18, and 33 for at least these further reasons.

c. THE REFERENCES TEACH AWAY FROM THE CLAIMED INVENTION

The rejection of claims 1, 18, 33, 35 and 36 should also withdrawn because at Lewis teaches away from aspects of the invention as claimed. Cited art that teaches away is evidence of nonobviousness. (MPEP 2145.) A prior art reference may be considered to teach away when a person of ordinary skill, upon reading the reference, would be led in a direction divergent from the path that the Appelants took. (In re Gurley, 27 F.3d at 553, 31 USPQ 2d at 1131 (Fed. Cir. 1994).)

Several times, Lewis expressly teaches away from "specifying, according to one or more rules, an error pattern based upon a combination of one or more error events in the storage area network" and "associating an error action to perform according to the one or more rules and in response to receiving the combination of one or more error

events of the error pattern" as recited in Claim 1. In fact, Lewis details operation of a rule-based reasoning (RBR) system and then proceeds to point out all its failures (Col. 2, lines 22-66 of Lewis).

In the background section, Lewis describes performing a "consult/implement/test" cycle with an expert and then reducing the methodology "in a rule or rules that the system can process" (Col. 2, lines 50-55). Subsequently, Lewis states that the resulting rule based system suffers from "brittleness" because it fails as soon as "it is presented with a problem for which it has no applicable rules" (Col. 2, lines 60-63). Rule based reasoning (RBR) also suffers from other various computing and logistical problems including a "knowledge acquisition bottleneck" when an engineer tries to change special rules and procedures that result in unforeseen situations (Col. 3, lines 1-5). Finally, Lewis declares that attempting to use rules to drive a system eventually results in "unwieldly, unpredictable, and unmaintainable" (Col. 3, lines 5-9) systems most likely to become obsolete.

In direct contrast with the rule based reasoning (RBR), Lewis extols the virtues of a case-based reasoning or CBR over RBR. Lewis specifically states that, "The case-based reasoning method of the invention represents fault resolution expertise in the form of cases, i.e. particular, specific, fault resolution scenarios stored in completed trouble tickets, rather than general rules as used in existing RBR systems" (Col. 6, lines 35-39 of Lewis).

This language in Lewis states quite strongly that using general rules as recited in claim 1 should be avoided. Clearly, a person of ordinary skill, upon reading the reference,

would be led in a direction divergent from the path that the Appelants took and use cased-based reasoning (CBR) rather than rules as cited in Claim 1 and elsewhere.

Accordingly, Appellants respectfully requests that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to claims 1, 18 & 33 for this further reason.

d. DEPENDENT CLAIMS 2-4, 6-11, 19-21, AND 23-26 ARE ALSO IN CONDITION FOR ALLOWANCE.

While claims 2-4, 6-11, 19-21, and 23-26 are allowable independently, they are also in condition for allowance by virtue of their dependence from claims 1 and 18 respectively.

B. CLAIM 12, CLAIM 27, CLAIM 34 AND CLAIM 36 ARE NONOBVIOUS OVER LEWIS IN VIEW OF REUTER AND FURTHER IN VIEW OF FERIDUN AND D'SOUZA.

a. NO TEACHING, SUGGESTION OR MOTIVATION TO COMBINE LEWIS WITH REUTER AND FURTHER IN VIEW OF FERIDUN AND D'SOUZA

Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. (MPEP 2143.01.) The suggestion or motivation to combine the reference teachings must be based on some logical reason apparent from positive, concrete evidence that justifies the combination. (In re Laskowski, 871 F.2d 115, 10 USPQ 2d 1397 (Fed. Cir. 1989).)

For at least the same reasons previously described, there is no reason to combine Lewis with Reuter.

Regarding Feridun, the Examiner summarily concludes that a person of ordinary skill in the art would be motivated "to use such an event correlator because Lewis and Reuter disclose a complete distributed environment in which events are used for fault diagnosis" (Page 9, paragraph 1 of Office Action dated May 10, 2007). It appears this conclusory statement justifying the combination may have been based upon *KSR International Co. v. Teleflex Inc.*, 550 U.S. --, 82 USPQ2d 1385 (2007). Indeed, the Examiner has stated that the "KSR ruling teaches that any person of ordinary skill would have looked to any relevant technology in searching for solutions that meet some need, and indeed, Examiner need only show that such a person *could* have looked to the relevant technology (underline emphasis added)" (Page 5, Item 15 of the Office Action dated November 20, 2007).

However, KSR is not some "silver bullet", the mere citation to which substitutes for the "positive, concrete evidence that justifies the combination". In fact, the Appellants note that "rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." (In re Kahn, 441 F.3d 977, 988 (CA Fed. 2006) cited with approval in KSR.) In this instance, the Examiner has merely made a conclusory statement without requisite articulated reasoning or with rational underpinning in support of the obviousness combinations and rejections.

Indeed, the Examiners conclusory statement is also not accurate as Lewis and Reuter describe two unrelated technologies – they do not both deal with "fault diagnosis"

as the Examiner has also asserted. As previously described, Lewis concerns a trouble ticket system that stores problem tickets related to network failures. To find a trouble ticket, Lewis relies upon case-based reasoning (CBR) over the use of rules as found in rule-based reasoning (RBR). Compare Reuters that does not deal with network failures or diagnosis at all but "page faults" in virtual storage. As previously defined and to those skilled in the art, "page faults" are not related to "network failures" in this context.

Concerning D'Souza, the Examiner merely repeats an overarching goal found in the abstract of D'Souza that states "maintaining a predefined acceptable fault tolerance level for a plurality of software modules implementing a software program running on a first plurality of computers coupled together in a cluster configuration in a first cluster in a clustered computer system" (Page 10, paragraph 1 of Office Action dated May 10, 2007). This conclusory and goal oriented statement copied directly from the Abstract of D'Souza clearly does include any "articulated reasoning" or "rational underpinning" for making the further combination of D'Souza with Lewis in view of Reuter and further in view of Feridun.

There is no teaching, suggestion or motivation in the prior art to combine the cited art to result in what is claimed, as is required. Consequently, Claim 12 is not obvious over Lewis in view of Reuter and further in view of Feridun and D'Souza for at least the reasons set forth above. By analogy independent claims 27 and 34 are also in condition for allowance for at least the same reasons.Accordingly, Appellant respectfully requests that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to claims 12, 27 and 34 for at least these reasons.

b. ALL CLAIM LIMITATIONS ARE NOT TAUGHT OR SUGGESTED BY THE CITED ART.

With regards to the prima facie case, it is a requirement that all the claim limitations must be taught or suggested by the prior art. MPEP § 2143.03 In re Royka , 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson , 424 F.2d 1382, 165 USPQ 494, 496 (CCPA 1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. In re Fine , 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

Contrary to the Examiner's assertion, Lewis alone or in combination with the cited art does not teach each and every feature of the invention as claimed. Lewis does not operate by "initializing a primary failure analysis module for processing error events and error actions" as recited in Claim 12.

Nonetheless, to satisfy this limitation in the office action of May 10, 2007, the Examiner merely states "a functioning process must have been initialized at some point" (Office Action mailed May 10, 2007, Page 2, Paragraph 4). Later the Examiner reasons "it is difficult to imagine how anything can come into existence without a starting point. Computer logic most certainly is initialized at some point with the relevant data, and this is most certainly the case in Lewis" (Office Action mailed November 20, 2007).

Even though these conclusory statements may appeal to the Examiner's common sense, there is nothing cited as a basis for the assumptions made. (Feature missing from references; improper to rely on basic knowledge and common sense) *In Re Zurko*, 258 F.3d 1379, 59 USPQ2d 1693 (Fed. Cir. 2001) *on remand from Dickinson v. Zurko,* 527 U.S. 150, 50 USPQ2d 1930 (1999) Even so, the Examiner's unsupported reasoning also

does not teach or suggest "initializing a primary failure analysis module for processing error events and error actions" as recited in claim 12.

As the Examiner has also noted, Lewis also does not teach or suggest "comparing, according to one or more rules, a temporal arrangement of the error events received against a set of error patterns loaded in the failure analysis module" as recited in amended claim 12. (Page 8 of the Office Action mailed May 10, 2007) Not only does Lewis not record error events or contemplate a SAN device, there is no teaching, suggestion or motivation to perform a temporal comparison of error events according to one or more rules. Indeed, Lewis operates on "cases" or "scenarios" and therefore cannot perform discrete temporal comparison of events for further analysis. (Col. 6, lines 36-39). Since Lewis teaches away from using general rules and even rule-based reasoning (RBR) (Col. 6, lines 35-39) then Lewis should not be combined with the use of correlation rules in Feridun.

Even if they were combined, Lewis with Reuter and further in view of Feridun does not teach or suggest the limitations as recited in claim 12. Indeed, in order to determine whether one of ordinary skill has good reason to combine known options depends on when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions within the person's technical grasp. *KSR Int'l Co. v. Teleflex, Inc.,* No. 04-1350, slip op. at 17 (U.S. April 30, 2007) Here, Feridun provides a correlation engine for processing correlator rules that may be programmed without any clearly finite limitations. (Col. 9, lines 1-67; Col. 10, lines 1-67.) Accordingly, it cannot be expected that one skilled in the art would try to perform a

comparison temporally in the fashion described in claim 12 as there is much more than a finite number of solutions.

For at least these reasons, independent claims 12, 27, 34 and 36 remain patentable over the cited art.

Accordingly, Appellant respectfully requests that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to claims 12, 27, 34 and 36 for at least these further reasons.

c. DEPENDENT CLAIMS 13-16 AND 28-32 ARE ALSO IN CONDITION FOR ALLOWANCE

Dependent claims 13-16 and 28-32 are not only independently patentable but also allowable by virtue of their dependency on independent claims 12 and 27 respectively.

d. THE REFERENCES TEACH AWAY FROM THE CLAIMED INVENTION

The rejection of claims 12, 27, 34 and 36 should also withdrawn because Lewis teaches away from aspects of the invention as claimed. Cited art that teaches away is evidence of nonobviousness. (MPEP 2145.) A prior art reference may be considered to teach away when a person of ordinary skill, upon reading the reference, would be led in a direction divergent from the path that the Appelants took. (In re Gurley, 27 F.3d at 553, 31 USPQ 2d at 1131 (Fed. Cir. 1994).)

As previously described, Lewis expressly teaches away from "comparing , according to one or more rules, a temporal arrangement of the error events received against a set of error patterns loaded in the primary failure analysis module" and "identifying the error pattern from the set of error patterns and error action corresponding

to the error pattern to perform in response to the comparison in the primary failure analysis module and the one or more rules" as recited in Claim 12. In fact, Lewis details operation of rules and/or a rule-based reasoning (RBR) system and then proceeds to point out all its failures (Col. 2, lines 22-66 of Lewis).

In the background section, Lewis describes performing a "consult/implement/test" cycle with an expert and then reducing the methodology "in a rule or rules that the system can process" (Col. 2, lines 50-55). Subsequently, Lewis states that the resulting rule based system suffers from "brittleness" because it fails as soon as "it is presented with a problem for which it has no applicable rules" (Col. 2, lines 60-63). Rule based reasoning (RBR) also suffers from other various computing and logistical problems including a "knowledge acquisition bottleneck" when an engineer tries to change special rules and procedures that result in unforeseen situations (Col. 3, lines 1-5). Finally, Lewis declares that attempting to use rules to drive a system eventually results in "unwieldly, unpredictable, and unmaintainable" (Col. 3, lines 5-9) systems most likely to become obsolete.

In direct contrast with the rule based reasoning (RBR), Lewis extols the virtues of a case-based reasoning or CBR over RBR. Lewis specifically states that, "The case-based reasoning method of the invention represents fault resolution expertise in the form of cases, i.e. particular, specific, fault resolution scenarios stored in completed trouble tickets, rather than general rules as used in existing RBR systems" (Col. 6, lines 35-39 of Lewis).

This language in Lewis states quite strongly that using general rules as recited in claim 12 should be avoided. Clearly, a person of ordinary skill, upon reading the

reference, would be led in a direction divergent from the path that the Appelants took and use cased-based reasoning (CBR) rather than rules as cited in Claim 12 and elsewhere.

Accordingly, Appellants respectfully requests that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to claims 1, 18, 33, 35 and 36 for this further reason.

C. CLAIM 35 IS NONOBVIOUS OVER LEWIS IN VIEW OF REUTER AND THE DEFINITION OF "THRESHOLD" BY THE IEEE AND "GRAPHICAL USER INTERFACE" BY THE MICROSOFT COMPUTER DICTIONARY.

a. ALL CLAIM LIMITATIONS ARE NOT TAUGHT OR SUGGESTED BY THE CITED ART.

With respect to independent claim 35, the Examiner indicated claim 35 was obvious under 35 U.S.C. § 103(a) over Lewis in view of Reuter and the definition of "threshold" by the IEEE and "graphical user interface" in view of Harper. However, as previously described, Lewis taken alone or together with Reuter does not concern "initializing a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy" as recited in claim 35.

It also follows that Lewis does not describe or even suggest "identifying one or more predetermined error actions and one or more error events associated with the storage area network" as recited in claim 35. Unfortunately, the office action dated May 10, 2007 cites Figure 4 from Lewis but does not consider the overall reference as a whole

and the actual underlying explaination provided in Lewis with respect to Figure 4 and elsewhere.

In particular, Lewis does not identify error events as recited in claim 1 but instead identifies "cases" using "case-based reasoning" or CBR. (Col. 6, lines 33-39). Lewis makes it a point that a "case" or "scenario" described by a trouble ticket is distinguishable over a combination of error actions and error events – this is a central concept to Lewis. (Col. 3, lines 11-20; Col. 6, lines 33-39.) Lewis very specifically identifies "cases" having "particular, specific, fault resolution scenarios stored in completed trouble tickets" and teaches away from identifying error actions and error events. (*Id.*) By operating on cases, Lewis attempts to resolve faults by operating with a different approach. Specifically, if a trouble-ticket matches exactly, Lewis immediately performs the resolution performed previously in the matching trouble ticket. (Col. 8, lines 55-60) In the event there is not an exact match, Lewis describes adapting the trouble-ticket using a parameterized adaptation module and other approaches. (Col. 8, lines 60-67; Col. 9, lines 1-67).

Reuter does not concern processing either cased-based reasoning or errors but page faults that occur on SAN. As previously described, there is nothing in Reuter that relates to managing errors let alone the limitations associated with the steps of "specifying an error pattern based upon a combination of one or more error events in the storage area network and according to one or more rules, presented through a graphical user interface with corresponding threshold values" and "associating an error action presented through the graphical user interface to perform according to one or more rules

and in response to receiving the combination of one or more error events of the error pattern that satisfy the threshold value requirements" as recited in claim 35. If and when a page fault occurs Reuter will either ignore a storage location or pass the error code to an application (Paragraph 27, lines 1-9 of Reuter.). Nothing in Reuter describes performing error actions as recited in claim 35.

Indeed, Lewis combined with Reuter does not teach or suggest "initializing...an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy" as recited in claim 35.

For at least these reasons, independent claim 35 as currently filed is in condition for allowance.

Accordingly, Appellant respectfully requests that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to claim 35 for at least these further reasons.

VIII. CONCLUSION

Appellant respectfully submits it has demonstrated that the cited references do not teach or suggest each and every element of the pending claims 1-4, 6-16, 18-21, 23-36 and that the rejections under 35 U.S.C. § 103(a) cannot be maintained.

For at least the reasons discussed above, Appellant submits that the pending claims are patentable. Accordingly, Appellant requests that the Board of Appeals reverse the Examiner's decisions regarding claims 1-4, 6-16, 18-21, 23-36.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.

Respectfully submitted,

Date: 4/20/2008

Leland Wiesner
Reg. No. 39,424

Attorneys for Appellant
Wiesner and Associates
366 Cambridge Ave.
Palo Alto, CA. 94306
Phone: (650) 853-1113
Fax: (650) 853-1114

CLAIMS APPENDIX A

What is claimed is:

1. (Previously presented) A method for configuring a storage virtualization controller to manage errors in a storage area network, comprising:

initializing a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy;

identifying one or more predetermined error actions and one or more error events associated with the storage area network;

specifying, according to one or more rules, an error pattern based upon a combination of one or more error events in the storage area network; and

associating an error action to perform according to the one or more rules and in response to receiving the combination of one or more error events of the error pattern.

2. (Original) The method of claim 1 further comprising loading the error pattern and associated error action into a failure analysis module.

3. (Previously presented) The method of claim 1 further comprising initializing the failure analysis module with the one or more predetermined error actions, one or more predetermined system error events and one or more predetermined input-output error events associated with the storage area network.

4. (Original) The method of claim 1 wherein the configuration and management is performed using a centralized failure analysis module.

5. (Canceled)

6. (Previously presented) The method of claim 1 wherein each of the one or more predetermined error actions describes a set of operations to accommodate the occurrence of the one or more system error events and one or more input-output error events.

7. (Original) The method of claim 1 wherein the one or more error events are selected from a set of error events including predetermined system error events and predetermined input-output error events.

8. (Original) The method of claim 7 wherein each of the one or more system error events occurs when an error event occurs corresponding to a module within the storage virtualization controller.

9. (Previously presented) The method of claim 1 wherein each of one or more input-output error events corresponds to a communication error between the storage virtualization controller and servers or storage elements in the storage area network.

10. (Original) The method of claim 1 wherein the error pattern and associated error actions are specified incrementally over time without recoding.

11. (Original) The method of claim 1 wherein the error pattern is generated automatically through a logging and analysis of past error events.

12. (Previously presented) A method of managing the occurrence of errors generated in a storage area network, comprising:

initializing a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy;

generating one or more error events responsive to the occurrence of one or more conditions of components being monitored in the storage area network;

receiving the one or more error events over a time interval for analysis in the primary failure analysis module;

comparing , according to one or more rules, a temporal arrangement of the error events received against a set of error patterns loaded in the primary failure analysis module; and

identifying the error pattern from the set of error patterns and error action corresponding to the error pattern to perform in response to the comparison in the primary failure analysis module and the one or more rules.

13. (Original) The method of claim 12 wherein the one or more error events are converted into error event codes by a set of monitor modules monitoring the components in the storage area network.

14. (Original) The method of claim 12 wherein the one or more error events are selected from a set of error events including predetermined system error events and predetermined input-output error events.

15. (Original) The method of claim 14 wherein each of the one or more system error events occurs when an error event occurs corresponding to a module within a storage virtualization controller.

16. (Previously presented) The method of claim 14 wherein each of the one or more predetermined input-output error events corresponds to a communication error between the storage virtualization controller and servers or storage elements in the storage area network.

17. (Canceled)

18. (Previously presented) An apparatus that configures a storage virtualization controller to manage errors in a storage area network, comprising:

a processor capable of executing instructions;

a memory containing instructions capable of execution on the processor that cause the processor to initialize a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy, identify one or more predetermined error actions and one or more error events associated with the storage area network, specify ,according to one or more rules, an error pattern based upon a combination of one or more error events in the storage area network and associate an error action to perform according to the one or more rules and in response to receiving the combination of one or more error events of the error pattern.

19. (Previously presented) The apparatus of claim 18 further comprising instructions in the memory when executed load the error pattern and associated error action into the failure analysis module in the memory.

20. (Previously presented) The apparatus of claim 18 further comprising instructions in the memory when executed initialize the failure analysis module with the one or more predetermined error actions, one or more predetermined system error events and one or more predetermined input-output error events associated with the storage area network.

21. (Original) The apparatus of claim 18 wherein the configuration and management is performed using a centralized failure analysis module.

22. (Canceled)

23. (Previously presented) The apparatus of claim 18 wherein each of the one or more predetermined error actions describes a set of operations to accommodate the occurrence of the one or more system error events and one or more input-output error events.

24. (Original) The apparatus of claim 18 wherein the one or more error events are selected from a set of error events including predetermined system error events and predetermined input-output error events.

25. (Original) The apparatus of claim 24 wherein each of the one or more system error events occurs when an error event occurs corresponding to a module within the storage virtualization controller.

26. (Previously presented) The apparatus of claim 18 wherein each of one or more input-output error events corresponds to a communication error between the storage virtualization controller and servers or storage elements in the storage area network.

27. (Previously presented) An apparatus for managing the occurrence of errors generated in a storage area network, comprising:

a processor capable of executing instructions;

a memory containing instructions when executed on the processor initialize a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy, generate one or more error events responsive to the occurrence of one or more conditions of components being monitored in the storage area network, receive the one or more error events over a time interval for analysis in the primary failure analysis module, compare, according to one or more rules,

a temporal arrangement of the error events received against a set of error patterns loaded in the primary failure analysis module and identify the error pattern from the set of error patterns and error action corresponding to the error pattern to perform in response to the comparison in the primary failure analysis module and the one or more rules.

28. (Original) The apparatus of claim 27 wherein the one or more error events are converted into error event codes by a set of monitor modules monitoring the components in the storage area network.

29. (Previously presented) The apparatus of claim 27 wherein the one or more error events are selected from a set of error events including predetermined system error events and predetermined input-output error events.

30. (Previously presented) The apparatus of claim 29 wherein each of the one or more system error events occurs when an error event occurs corresponding to a module within the storage virtualization controller.

31. (Previously presented) The apparatus of claim 29 wherein each of the one or more input-output error events corresponds to a communication error between the storage virtualization controller and servers or storage elements in the storage area network.

32. (Previously presented) The apparatus of claim 27 wherein the failure analysis module receiving the one or more error events is configured as a primary failure analysis module for processing error events and one or more additional alternate failure analysis modules are configured as backups to the primary failure analysis module and the alternate failure analysis module to facilitate high-availability and redundancy.

33. (Previously presented) An apparatus for configuring a storage virtualization controller to manage errors in a storage area network, comprising:

means for initializing a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy;

means for identifying one or more predetermined error actions and one or more error events associated with the storage area network;

means for specifying, according to one or more rules, an error pattern based upon a combination of one or more error events in the storage area network; and

means for associating an error action to perform according to the one or more rules and in response to receiving the combination of one or more error events of the error pattern.

34. (Previously presented) An apparatus for managing the occurrence of errors generated in a storage area network, comprising:

means for initializing a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy;

means for generating one or more error events responsive to the occurrence of one or more conditions of components being monitored in the storage area network;

means for receiving the one or more error events over a time interval for analysis in the primary failure analysis module;

means for comparing , according to one or more rules, a temporal arrangement of the error events received against a set of error patterns loaded in the primary failure analysis module; and

means for identifying the error pattern from the set of error patterns and error action corresponding to the error pattern to perform in response to the comparison in the primary failure analysis module and the one or more rules.

35. (Previously presented) A method for configuring a storage virtualization controller to manage errors in storage area network, comprising:

initializing a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy;

identifying one or more predetermined error actions and one or more error events associated with the storage area network;

specifying an error pattern based upon a combination of one or more error events in the storage area network and according to one or more rules, presented through a graphical user interface with corresponding threshold values; and

associating an error action presented through the graphical user interface to perform according to one or more rules and in response to receiving the combination of one or more error events of the error pattern that satisfy the threshold value requirements.

36. (Previously presented) A method for configuring a storage virtualization controller to manage errors in a storage area network, comprising:

initializing a primary failure analysis module for processing error events and error actions and an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy;

generating , according to one or more rules, one or more error patterns automatically through logging of error events and analysis of the error events occurring in the storage area network over a time interval;

suggesting that the one or more error patterns generated from the analysis receive at least one error action to be performed according to the one or more rules and in the event the one or more error patterns occur on the storage area network; and

associating an error action to perform in response to each of the suggested one or more error patterns generated from the analysis.